

VŠB – Technická univerzita Ostrava  
Fakulta strojní  
Katedra automatizační techniky a řízení

**Využití open-source hardware pro zpracování veřejných dat**

**Display of Public Data Using Open-source Hardware**

Student:

Bc. Adam Czudek

Vedoucí diplomové práce:

Ing. Pavel Smutný, Ph.D.

Ostrava 2016

## Zadání diplomové práce

Student: **Bc. Adam Czudek**  
Studijní program: N2301 Strojní inženýrství  
Studijní obor: 3902T004 Automatické řízení a inženýrská informatika  
Téma: **Využití open-source hardware pro zpracování veřejných dat**  
**Display of Public Data Using Open-source Hardware**  
Jazyk vypracování: čeština

### Zásady pro vypracování:

1. Popište princip a charakteristiky open-source hardware a proveďte analýzu zobrazování dat z veřejných rozhraní pro programování aplikací (API).
2. Proveďte analýzu a srovnání realizovaných řešení úloh zobrazování dat z veřejných API.
3. Navrhněte vhodné hardwarové prostředky pro vlastní realizaci úlohy zobrazování dat z veřejných API.
4. Navrhněte a zpracujte vlastní úlohu s využitím zobrazování veřejných dat.
5. Zhodnoťte dosažené výsledky a navrhněte směry dalšího řešení.

### Seznam doporučené odborné literatury:

MARGOLIS, Michael. Arduino cookbook. 2nd ed. Sebastopol, Calif.: O'Reilly, c2012, 699 p. ISBN 1449313876.  
MELGAR, Enrique Ramos, CASTRO DÍEZ, Ciriaco & JAWORSKI, Przemek. Arduino and Kinect projects: Design, build, blow their minds. New York: Distributed to the book trade worldwide by Springer Science+Business Media, c2012, 393 p. ISBN 9781430241683.  
LAVIN, Peter. PHP - objektově orientované: koncepty, techniky a kód. 1. vyd. Praha: Grada, 2009, 211 s. ISBN 9788024721378.  
FACEBOOK.COM. Facebook for Developers [online]. 2014 [cit. 2014-03-07]. Dostupné z: <http://developers.facebook.com>

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Pavel Smutný, Ph.D.**

Datum zadání: 11.12.2015

Datum odevzdání: 16.05.2016



doc. Ing. Renata Wagnerová, Ph.D.  
vedoucí katedry



doc. Ing. Ivo Hlavatý, Ph.D.  
děkan fakulty

### **Místopřísežné prohlášení studenta**

Prohlašuji, že jsem celou diplomovou práci včetně příloh vypracoval samostatně pod vedením vedoucího diplomové práce a uvedl jsem všechny použité podklady a literaturu.

V Ostravě, dne 16.5.2016

.....

podpis studenta

## **Prohlašuji, že**

- jsem byl seznámen s tím, že na moji diplomovou práci se plně vztahuje zákon č. 121/2000 Sb., autorský zákon, zejména § 35 – užití díla v rámci občanských a náboženských obřadů, v rámci školních představení a užití díla školního a § 60 – školní dílo.
- беру на ве́доміі, že Vysoká škola báňská – Technická univerzita Ostrava (dále jen „VŠB-TUO“) má právo nevýdělečně ke své vnitřní potřebě diplomovou práci užít (§ 35 odst. 3).
- souhlasím s tím, že diplomová práce bude v elektronické podobě uložena v Ústřední knihovně VŠB-TUO k nahlédnutí a jeden výtisk bude uložen u vedoucího diplomové práce. Souhlasím s tím, že údaje o kvalifikační práci budou zveřejněny v informačním systému VŠB-TUO.
- bylo sjednáno, že s VŠB-TUO, v případě zájmu z její strany, uzavřu licenční smlouvu s oprávněním užít dílo v rozsahu § 12 odst. 4 autorského zákona.
- bylo sjednáno, že užít dílo – diplomovou práci nebo poskytnout licenci k jejímu využití mohou jen se souhlasem VŠB-TUO, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly VŠB-TUO na vytvoření díla vynaloženy (až do jejich skutečné výše).
- беру на ве́доміі, že odevzdáním své práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, bez ohledu na výsledek její obhajoby.

V Ostravě, dne 16.5.2016

.....

podpis studenta

Jméno a příjmení autora práce: Adam Czudek

Adresa trvalého pobytu autora práce: Okružní 452, Orlová-Lutyně

# ANOTACE DIPLOMOVÉ PRÁCE

CZUDEK, A. *Využití open-source hardware pro zpracování veřejných dat: diplomová práce*. Ostrava : VŠB – Technická univerzita Ostrava, Fakulta strojní, Katedra automatizační techniky a řízení, 2016, 50 s. Vedoucí práce: Smutný, P.

Diplomová práce se zabývá návrhem a realizací vlastního řešení pro zpracování a zobrazování dat z veřejných webových zdrojů. V teoretické části jsou přiblíženy základní informace týkající se open-source hardware. Popsány jsou možnosti a význam open-source hardware v oblasti informačních technologií. Další kapitola je věnovaná webovým službám a API. Důraz je kladen na data ze sociálních sítí a volně dostupná data z veřejných zdrojů. Praktická část práce popisuje návrh a postup při realizaci úloh pro zpracování a zobrazování veřejných dat. Na základě vybraných komponent byl nakonec sestaven výsledný model displeje, založený na programovací desce Arduino Yún.

# ANNOTATION OF MASTER THESIS

CZUDEK, A. *Display of Public Data Using Open-source Hardware : Master thesis*. Ostrava : VŠB – Technical University of Ostrava, Faculty of Mechanical Engineering, Department of Control Systems and Instrumentation, 2016, 50 p. Thesis head: Smutný, P.

This thesis describes the design and implementation of custom solution for processing and displaying data from public web sources. The theoretical part presents basic information about open-source hardware. There are described the possibilities and the importance of open-source hardware in the field of information technology. Next chapter is devoted to web services and APIs. Emphasis is placed on data from social networks and data freely available from public sources. The practical part describes the design and process of implementation of tasks for processing and displaying public data. According to selected components was built the final model of the display based on Arduino Yun programming board.

# OBSAH

SEZNAM POUŽITÝCH ZNAČEK A ZKRATEK .....	7
1 ÚVOD .....	8
2 OPEN-SOURCE HARDWARE .....	9
2.1 Význam a charakteristika.....	9
2.2 Příklady open-source hardware .....	11
3 VEŘEJNÁ DATA A API.....	14
3.1 Webové služby a API .....	14
3.2 Formáty a protokoly.....	16
3.3 Služby a platformy pro zpracování API .....	17
4 REALIZOVANÉ ÚLOHY ZOBRAZOVÁNÍ DAT .....	19
4.1 Využití dat ze sociálních sítí.....	19
4.2 Zobrazování dat z webových služeb .....	21
5 NÁVRH KOMPONENTŮ PRO VLASTNÍ MODEL .....	23
5.1 Desky a moduly Arduino .....	23
5.2 Výběr hardwarových prostředků pro model displeje.....	26
6 ZPRACOVÁNÍ ÚLOH ZOBRAZOVÁNÍ DAT .....	29
6.1 Sestavení modelu displeje.....	30
6.2 Připojení a konfigurace desky Arduino Yún .....	33
6.3 Realizace úloh pro zpracování a zobrazování dat.....	35
7 ZÁVĚR .....	47
SEZNAM POUŽITÝCH ZDROJŮ .....	48

---

## SEZNAM POUŽITÝCH ZNAČEK A ZKRATEK

API	Application programming interface
CD	Compact disc
FB	Facebook
GNU	GNU's not unix
HTML	Hapertext markup language
ID	Identification
IoT	Internet of Things
JSON	Javascript object notation
LCD	Liquid crystal display
LED	Light-emitting diode
OSHW	Open-source hardware
OSS	Open-source software
OUT	Output
PC	Personal computer
PHP	Hypertext preprocessor
RSS	Rich site summary
SD	Secure digital
SRAM	Static random access memory
USB	Universal serial bus
UTP	Unshielded twisted pair
Wi-Fi	Wireless fidelity
XML	Extensible markup language

# 1 ÚVOD

Open-source hardware zažívá v posledních letech velký rozmach. Je však stále spíše na začátku cesty a na pozadí open-source software, který hraje ve světě informačních technologií již delší dobu významnou roli. Důvodem je především skutečnost, že vývoj hardwaru je náročnější než vývoj volně šiřitelného softwaru a v případě hardwaru se také jedná o vyšší výrobní náklady a nutnost dalších prostředků pro další inovace. Velkým rozdílem je zejména to, že open source software může vytvořit prakticky kdokoli a lze jej jednoduše distribuovat prostřednictvím internetu. V případě hardwaru jsou nároky často mnohem vyšší. Významným faktorem je také konkurence v podobě uzavřených komerčních řešení. V současné době je tak relativně obtížné vyvinout jakékoliv nové zařízení, u něhož by neexistovala potenciální hrozba kolize s patenty velkých výrobců.

Termínem, který je spojen mimo jiné s open-source hardware je tzv. Internet věcí (IoT). Jedná se o moderní trend v oblasti komunikace a kontroly předmětů běžného využití mezi sebou nebo s člověkem, prostřednictvím technologií bezdrátového přenosu dat a internetu. Internet věcí se tedy ve zkratce dá chápat jako síť různých zařízení, se kterými se každý z nás setkává v běžném životě. Hlavním cílem je aby všechna zařízení měla svou vlastní identitu a schopnost inteligentně komunikovat prostřednictvím různých rozhraní. Účelem a vizí do budoucna je vzájemná komunikace jednotlivých zařízení a sdílení co největšího množství informací.

Příslib do příštích let je v tomto ohledu opravdu velký. Nové inteligentní sítě již nebudou plnit jenom základní funkce, jako je propojení uživatelů mezi sebou, ale v podstatě vznikne propojení i s okolním prostředím. To bude zajištěno množstvím zařízení, které budou schopny poskytovat informace o prostředí, ve kterém se nacházejí. Dynamika vývoje v tomto směru se významně projevuje již dnes a souvisí především s velkou dostupností a nízkou pořizovací cenou těchto zařízení.

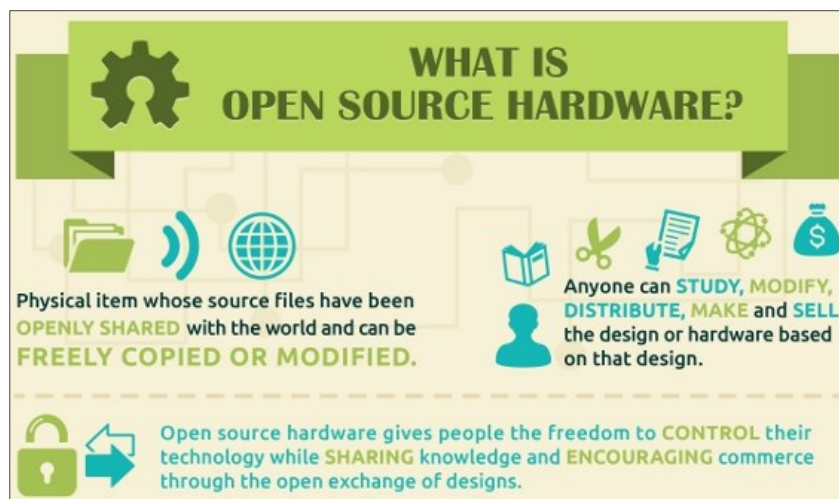


## 2 OPEN-SOURCE HARDWARE

Termínem open-source hardware (OSHW) se rozumí licence umožňující volné šíření návrhů elektronických zařízení. OSHW významově vychází z open-source software (OSS). U hardwaru se ovšem jedná o fyzická zařízení, licence je tak částečně upravena. Stanoví mimo jiné to, že volně šířená zařízení by měla být veřejně přístupná, aby je měl možnost kdokoliv studovat, upravovat a dále prodávat. Je běžnou praxí, že ke každému výrobku je veřejně dostupná dokumentace, návrh a schéma plošného spoje [RUBEŠ, P. 2013].

### 2.1 Význam a charakteristika

Uživatel má možnost zakoupené zařízení studovat a upravovat podle vlastních potřeb. Takto lze relativně snadným a levným způsobem vytvořit prototyp. Důležitá věcí je neustálá inovace, díky se v čele drží takové firmy, které mají snahu svoje zařízení posunout dále. Velký význam v tomto odvětví mají komunity lidí, které jsou inspirovány určitým OSHW zařízením. Tyto skupiny daná zařízení upravují a modifikují. Takovým způsobem vzniká velké množství variací původního návrhu [RUBEŠ, P. 2013].



Obr. 1 Základní rysy open-source hardware [OSHWA. 2013]

Struktura konkrétního konfigurovatelného systému se vyznačuje základními logickými prvky, které jsou navzájem propojeny sítí spojů a programovatelných spínačů. Prvky jsou dále hierarchicky sdruženy do bloků. V systému jsou kromě základních bloků obsaženy i moduly, které slouží pro implementaci složitějších variant. Funkce jak jednotlivých

elementárních prvků, tak i složitějších bloků jsou programovatelné. Konfigurace prvků a spínačů pro daný systém je uložena v různých typech paměťových buněk. Nejčastěji jsou využívané systémy s konfigurací ve statické paměti (SRAM) [ŠŤASTNÝ, J. 2011].

Obecným významem termínu open source je ve volném překladu otevřený kód. Autor zhotoví počítačový program nebo jeho část a rozhodne se, zda svůj výtvor zpřístupní ostatním. Většinou se jedná o neomezený a bezplatný přístup k takovému zdroji. Kdokoliv má pak možnost si stáhnout otevřený kód, dále jej upravovat, ověřit a publikovat. Výsledek své práce může potom prodávat. Podmínkou je ovšem zpřístupnění zdrojového kódu opět ostatním potenciálním uživatelům.

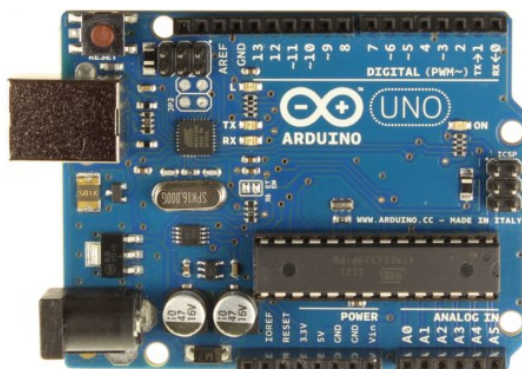
Z obchodního hlediska to často nabývá dvojí význam a může to stanovit výhodu jak i problém. V určitých případech se nedá dodržet licenci z bezpečnostních důvodů, v jiných zase kvůli konkurenčním hrozbám. Velká část běžných uživatelů má ovšem ve zvyku kupovat software v konečné podobě, protože většinou nedisponuje takovými znalostmi, aby mohl ladit a kompilovat kód. Ze strany vývojářů, kteří nabízejí hotové aplikace, je přidanou hodnotou jejich plná podpora. Mezi výhody otevřeného softwaru patří zejména zajištění vzájemné kompatibility aplikací, formátů apod. Typickým příkladem svobodné licence je GNU (General public licence).

Pokud se jedná o počítačový hardware, je zde situace odlišná. Výroba takových komponentů, jako je třeba základní deska nebo rozšiřující karty, je většinou otázkou vyšší ceny. Z tohoto důvodu nemůže být sama o sobě nabízena zdarma. Co k dispozici být může, je volně dostupná kompletní dokumentace, díky které je umožněna vlastní výroba konkrétního zařízení. Kdokoliv si potom může bez placení poplatků za licence, zařízení vyrobit nebo modifikovat a předat dál. V případě daných licenčních podmínek je také možnost provést změny v dokumentaci a bez omezení je předat veřejně dál. Mezi mnoha výhodami je zde také i jedna nevýhoda. Tou je nemožnost plného nároku na celé řešení. Konkrétní kód nebo obvody bývají upravovány ve srovnání s prací někoho jiného. Tím pádem je zde možnost, že v závislosti na licenci může existovat povinnost, která stanoví stejné podmínky pro ostatní. Z pohledu licencí to bývá v případě volně šířitelného hardware různé. Je třeba zmínit zejména open hardware licence. Klasickým příkladem svobodného hardwaru jsou zejména miniaturní počítače, různé víceúčelové moduly nebo 3D tiskárny [POLÁK, M. 2013].

## 2.2 Příklady open-source hardware

### ▪ Arduino

Arduino je typickým příkladem OSHW. Autorem této platformy je italská firma Smart Projects, která je podporovaná širokou komunitou uživatelů po celém světě. Arduino také poprvé stanovilo OSHW licenci a postupně se tato platforma velice rychle rozšířila mezi laiky i odborníky. Výhodou je zejména možnost rychlého vytvoření prototypu pro svůj produkt. K velkému nárůstu popularity došlo v roce 2010, kdy se za tento rok prodalo celkem 200 000 kusů těchto produktů. Arduino ve svých zařízeních používá mikroprocesory ATmega od firmy Atmel. U standardního mikroprocesoru ATmega je pro programování potřeba použít speciální programátor. Díky tomu, že se na desce nachází předinstalovaný ovladač, se mikroprocesor programuje přes sériovou linku (USB) přímo z vývojového prostředí. Tvorba kódu je realizována pomocí vyššího programovacího jazyka. Tento jazyk je založen na platformě Wiring. Tímto se významně sníží počáteční investice a dochází ke zjednodušení celého procesu programování [RUBEŠ, P. 2013].



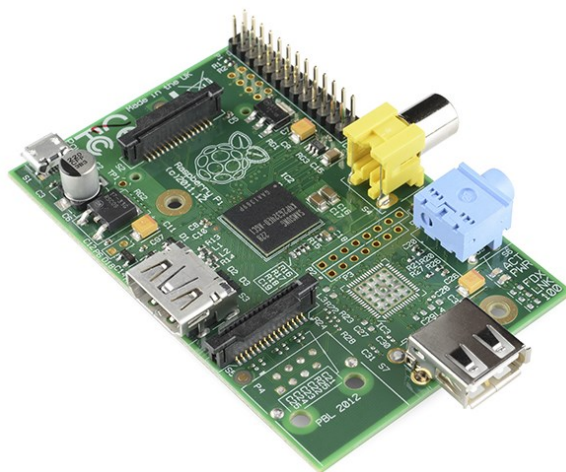
**Obr. 2 Arduino Uno [ARDUINO ÚVOD. 2013]**

### ▪ Raspberry Pi

Raspberry Pi je miniaturní počítač o velikosti kreditní karty. Má velmi široké možnosti připojení, protože lze k němu zapojit monitor, klávesnici či jiné periferie jako v případě klasického PC. Jedná se o relativně výkonné malé zařízení, které může mít spoustu různých variant využití. Je schopné zvládnout většinu práce, která se běžně provádí na klasickém PC. Konkrétně to může být práce s tabulkami, editace textu či jednoduché hry. Jeho velkou předností je především velmi malá spotřeba, a díky tomu je ideálním řešením na neustále běžící aplikace, jako je provoz datových úložišť, hostování webových stránek, virtuálních sítí apod.

Dodáván může být pouze jako samotné zařízení bez přídatných periférií. Raspberry Pi pro náběh systému a dlouhodobé skladování používá SD kartu a napájení je realizováno pomocí micro USB.

Původní myšlenka na levný počítač se zrodila v roce 2006 ve Velké Británii. Raspberry Pi nebyl na začátku zamýšlen jako komerční projekt, ale šlo o neziskovou dobročinnou nadaci, jejíž cílem bylo dát dětem k dispozici jednoduchý a dostupný prostředek, díky kterému se může zefektivnit výuka programování [HORÁK, M. 2012].

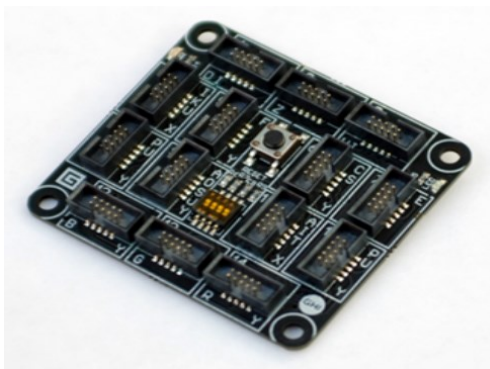


**Obr. 3 Raspberry Pi [RASPBERRY PI. 2014]**

#### ▪ .NET Gadgeteer

.NET Gadgeteer je další open source HW a SW platforma, vyvinutá společností Microsoft. Hlavním cílem je snaha o jednoduché vytváření různých zařízení, které mohou být využity k ovládání .NET Micro Framework. Jedná se v podstatě o modulární stavebnici, která otevírá široké spektrum využití především pro školy. Systém je navržen v takovém smyslu, aby mohl být co nejdostupnější pro začátečníky. Konektory na řídicí desce a jednotlivých modulech bývají standardně označeny písmeny. Tímto je jasné, co se má kam připojit a ulehčuje to orientaci.

Prvním komerčně dostupným zařízením podporujícím specifikaci .NET Gadgeteer byl FEZ Spider od GHI electronics. Jiný typ desky je FEZ Hydra. K těmto deskám je dostupné velké množství nejrozličnějších modulů, které se připojují pomocí standardizovaného konektoru. Velkým plusem je také podpora ze strany MS Visual Studio [BECHYNSKÝ, Š. 2012].



**Obr. 4 .NET Gadgeteer**  
**[MICROSOFT .NET GADGETEER. 2011]**

#### ▪ Intel Edison

Jedná se o nový model miniaturního počítačového modulu. Je osazen moderním pokročilým procesorem s 22 nm architekturou, tvořeným dvěma výkonnými jádry taktovanými na 500 Mhz. Navíc je integrováno ještě jedno jádro s architekturou Quark určené pro zpracování méně náročných úloh. Modul obsahuje mimo jiné integrovaný bezdrátový adaptér Broadcom, který umožňuje připojení přes Bluetooth a Wi-Fi nejnovějších standardů. Napájení je řešeno prostřednictvím 70pinového konektoru, který zároveň slouží jako výstup pro všechna rozhraní z procesoru. Celý modul běží pod operačním systémem Linux a i přes svůj velký výkon a výbavu má rozměry o velikosti běžné paměťové karty formátu SD.

Možnosti využití tohoto modulu jsou i s ohledem na velký potenciál do budoucna opravdu široké. Uplatnění najde především jako základní komponenta pro různé produkty v rámci Internetu věcí nebo nositelné elektroniky. Společnost Intel má navíc v plánu speciální obchod s aplikacemi pro podporu vývoje [OLŠAN, J. 2014].



**Obr. 5 Intel Edison [HOLLISTER, S. 2014]**

## 3 VEŘEJNÁ DATA A API

API je zkratka znamenající Application Programming Interface. Jde o termín softwarového inženýrství, který se dá v češtině interpretovat jako aplikační programové rozhraní. API je chápáno jako soubor funkcí a procedur konkrétní knihovny, aplikace či operačního systému, které mají možnost programátoři využívat namísto jejich programování. Jde o abstrakci funkcionality pro základní příkazy, které je možno zakomponovat a volat z jiných programů.

### 3.1 Webové služby a API

Webová služba stanoví souhrnné označení pro sadu technologií, které umožňují komunikaci mezi aplikacemi. Webové služby jsou v podstatě systémem pro podporu spolupráce mezi počítači v síti. Jedná se tedy o komunikaci mezi dvěma počítači, při níž jeden z nich plní funkci poskytovatele webové služby a druhý se chová jako klient. Poskytovatel služby zprostředkovává data určitým způsobem na síti. Na druhé straně je úkolem klienta zjistit adresu služby (vyhledáním v registru), stažení popisu služby a poté je schopen službu využívat. Webová služba je vlastně komponenta, která nabízí jednu, jasně definovanou službu, ať se jedná o převod měny, překlad textu nebo vyhledání řetězce přes vyhledávač.

#### ▪ Význam a druhy webových API

Velká část webových služeb poskytuje API pro vývojáře. Nejde jenom o služby pro zpracování textu, obrázků apod., ale v další řadě také o poskytování dat. Poskytovatelé webových služeb mají možnost ušetřit na testování své aplikace. Vývojáři využívající API v případě náhlých komplikací s implementací mohou využívat fóra s danou problematikou, nebo mohou přímo kontaktovat poskytovatele. Pokud se chyba objevila na straně koncového uživatele, může prostředník (vývojář) tuto záležitost poskytovateli API podrobně popsat. Poskytovatelé služeb mají také možnost se rozhodnout, zda do nějaké míry poskytovat své API zdarma. V případě, že klient projeví zájem o lepší služby, má možnost si objednat tzv. prémiový účet [HLAVÁČEK, J. 2011].

Mezi často se vyskytující druhy služeb patří tzv. mashupy. Jsou to webové služby, které jsou složeny kombinací dvou a více API. Klasickým příkladem mohou být hudební mashupy, které využívají propojení hudebních aplikací (Last.fm) se sociálními sítěmi (Facebook, Twitter) a mohou být dále napojeny na prodejce vstupenek.

### **Sociální síť**

Nejvíce rozšířená sociální síť Facebook využívá pro pochopení a popis všech objektů tzv. sociální graf. Každý objekt stanoví vrchol a hrany představují vzájemné vztahy mezi objekty. Objekt uživatel tak může mít vztah "like" ke stránce, "friend" k jinému objektu uživatel apod. Sociální graf ovšem nemusí končit na hranici Facebooku. Uživateli se může "líbit" cokoliv, na kterémkoliv místě na síti.

Další známou a oblíbenou sociální sítí je Twitter. Jde v podstatě o informační síť a komunikační službu založenou na krátkých zprávách. Platforma Twitteru umožňuje přístup k souboru dat prostřednictvím API, pomocí kterých mohou vývojáři vytvářet nebo dále rozšiřovat aplikace na základě dat Twitteru. Platforma se neustále rozvíjí, a proto vytvoření aplikace vyžaduje průběžné aktualizace. K dispozici je zde několik různých druhů nástrojů [DRYJÁK, T. 2012].

### **Poštovní služby**

V rámci poštovních služeb má velké zastoupení služba Gmail od společnosti Google. Umožňuje prostřednictvím API přístup k uživatelským schránkám a k základním operacím, jako je čtení příchozích zpráv, odesílání zpráv nebo vyhledávání konkrétních zpráv a vláken. Data jsou dostupná ve formátu JSON.

### **Meteorologické weby**

Služeb a webů věnovaných informacím o počasí existuje celá řada. Jednou z nich je služba OpenWeatherMap. Jedná se o web s volně dostupnými daty o počasí z celého světa. K dispozici je více než 200 tisíc měst a míst, jejichž informace jsou průběžně aktualizovány na základě dat a modelů z meteorologických stanic. Prostřednictvím rozhraní API jsou data dostupná ve formátech XML, JSON nebo HTML.

## 3.2 Formáty a protokoly

### ▪ HTML

HTML je základní značkovací jazyk pro publikaci libovolných dat v rámci sítě Internet. Je původně odvozen od značkovacího jazyka SGML (ISO 8879). Význam jednotlivých značek specifikovaný a předem daný. Tento jazyk neumožňuje, na rozdíl od jazyka XML, rozšíření o nové značky.

### ▪ XML

XML je jednoduchý značkovací jazyk s širokými možnostmi pro další rozšíření. Je specifikovaný konsorciem W3C a zejména díky jednoduchosti a také rozšiřitelnosti tohoto jazyka vzniklo velké množství implementací. Tento jazyk je určen především pro výměnu dat mezi aplikacemi. Dokumenty, které jsou vytvořené v tomto jazyku se orientují na strukturu přenášených dat a nezabývají se přímo vizuální stránkou dat. Základním prvkem pro tento značkovací jazyk byl jazyk SGML (ISO 8879). XML je běžně využíván pro prezentaci a výměnu informací jako jsou např. metadata.

### ▪ JavaScript

JavaScript je skriptovací jazyk, díky němuž lze jednoduchým způsobem oživit statické dokumenty HTML. Je schopný vytvářet objektově orientované skripty využívající událostí objektů jazyka HTML. Principem JavaScriptu je možnost spouštění kódu staženého z webového serveru na klientském počítači.

### ▪ JSON

Jedná se o odlehčený formát pro výměnu dat. Je čitelný a dobře zapisovatelný člověkem a lze jej snadno analyzovat a generovat také strojově. Je založen na podmnožině programovacího jazyka JavaScript, Standard ECMA-262.

JSON je nezávislý textový formát, který využívá konvence dobře známé rodiny C (C, C++, C#, Java, Python atd). Díky tomu je JSON pro výměnu dat velmi vhodným prostředkem [SOCHOROVÁ, M. 2008].

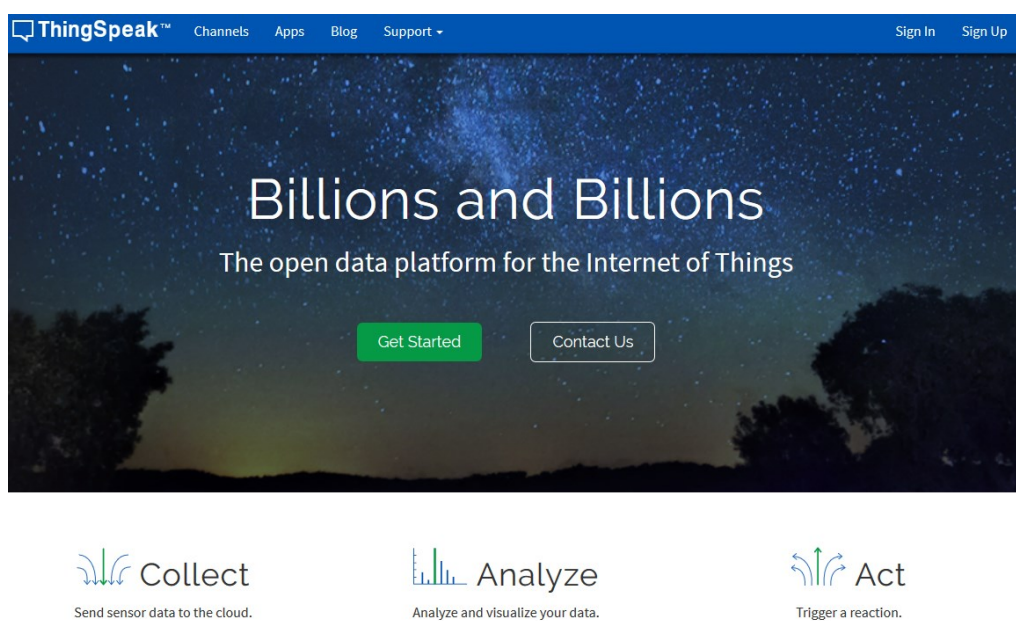


### 3.3 Služby a platformy pro zpracování API

Zpracování dat v rámci API webových služeb a sítí řeší mimo jiné také specializované služby, které usnadňují proces čtení a sběru dat. Tyto služby disponují často pokročilými nástroji jak pro získávání veřejných dat, tak pro sdílení vlastních dat.

- **ThingSpeak**

Platforma ThingSpeak využívá různorodé prvky IoT. Umožňuje získávání a ukládání dat ze síťových zařízení a jejich prezentaci ve formě grafů a jiných objektů. Uživatel je díky různým nástrojům schopen získat a následně zpracovat velké množství veřejných dat.

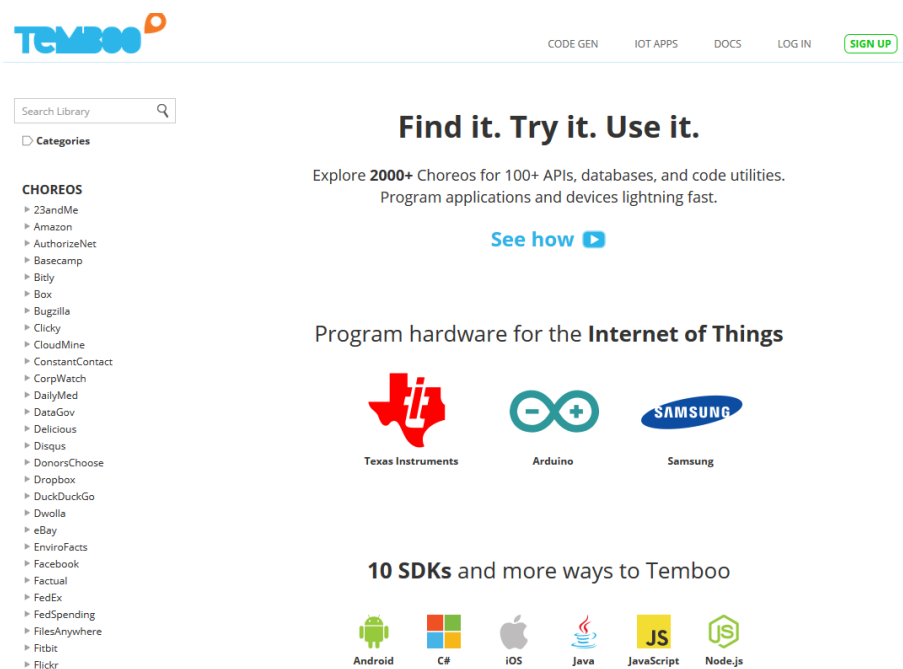


**Obr. 6 Výchozí stránka platformy ThingSpeak**

- **Temboo**

Tato pokročilá služba pro zpracování veřejných dat míří především na sociální sítě a informační weby. Disponuje stále se rozrůstající knihovnou s dostupnými zdroji webových API.

Velmi užitečnou funkcí je především nástroj pro generování výstupních dat ve formě zdrojového kódu pro programovací prostředí Arduino IDE. Takto připravený kód je možné dále upravovat a modifikovat pro vlastní potřeby.



**Obr. 7** Základní stránka služby Temboo

## 4 REALIZOVANÉ ÚLOHY ZOBRAZOVÁNÍ DAT

Díky rychlému vývoji a šíření webových služeb je v dnešní době dostupné poměrně velké množství veřejných dat. Tato otevřená data jsou na internetu volně přístupná ve strojově čitelné podobě v rámci API daného serveru, které představuje rozhraní webové služby.

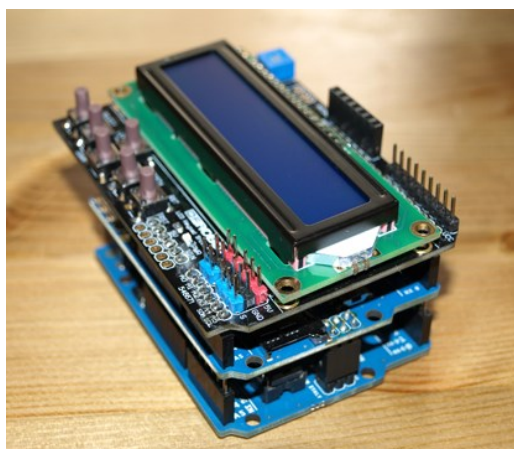
### 4.1 Využití dat ze sociálních sítí

Sociální sítě poskytují mnoho dat, které lze použít pro další zpracování. V následující části jsem uvedl úlohy, které využívají data ze sociální sítě Facebook.

První příklad popisuje realizaci jednoduchého zařízení pro zobrazování počtu fanoušků na profilovém účtu. Úloha je řešena použitím Arduino modulů a znakového LCD displeje. Napájení modulové desky je realizováno přes USB kabel a síťová komunikace probíhá prostřednictvím Ethernet modulu připojeného síťovým UTP kabelem k routeru.

Komponenty:

- Arduino Uno (základní modulová deska)
- Arduino Ethernet Shield (síťový modul)
- Arduino LCD Keypad Shield (znakový displej s tlačítky)
- USB kabel (typ A-B)
- UTP kabel (RJ45)



**Obr. 8 Displej připojený k Arduino modulům  
[FACEBOOK LIKE BOX. 2013]**

Zobrazení požadovaných dat je realizováno pomocí PHP skriptu vytvořeném v prostředí programovacího softwaru Arduino. V příkladu jsou použity knihovny Ethernet a String, které jsou součástí Arduino jádra.

```
30 #include <SPI.h>
31 #include <Ethernet.h>
32 #include <LiquidCrystal.h>
33
34 LiquidCrystal lcd( 8, 9, 4, 5, 6, 7 );
35
36 // Enter a MAC address and IP address for your controller below.
37 // The IP address will be dependent on your local network:
38 byte mac[] = {
39   0x00, 0xAA, 0xBB, 0xCC, 0xDE, 0x01 };
40 IPAddress ip(192,168,1,20);
41
42 // initialize the library instance:
43 EthernetClient client;
44
45 const unsigned long requestInterval = 12000; // delay between requests
46
47 char serverName[] = "api-read.facebook.com"; // facebook URL
48
49 boolean requested; // whether you've made a request since connecting
50 unsigned long lastAttemptTime = 0; // last time you connected to the server, in milliseconds
51
52 String currentLine = ""; // string to hold the text from server
53 String fbcount = ""; // string to hold the tweet
54 boolean readingFbcount = false; // if you're currently reading the tweet
55
```

**Obr. 9 Část kódu v prostředí softwaru Arduino  
[FACEBOOK LIKE BOX. 2013]**



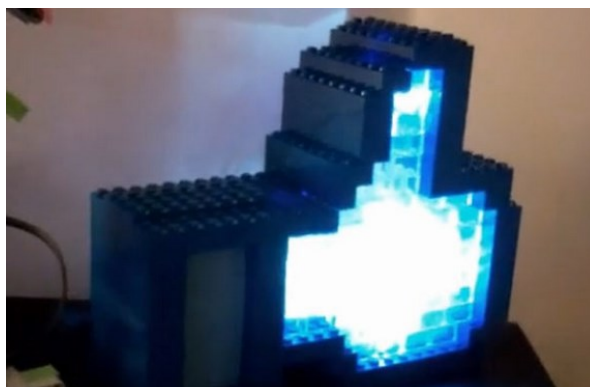
**Obr. 10 Zobrazení textu a výsledného počtu fanoušků  
[FACEBOOK LIKE BOX. 2013]**

Pokročilejším řešením, které předchází úlohu posouvá dále je komerční produkt s názvem Fliike. Jedná se o mechanické počítadlo, uvnitř kterého je deska Arduino s Wi-Fi modulem. K desce jsou připojeny motorky pro ovládání otočných bloků, které zobrazují jednotlivá čísla.



**Obr. 11 Mechanické počítadlo Flike [SMIIRL. 2014]**

Jiný způsob řešení představuje blikající model palce postavený z kostek Lega. Tento model symbolizuje populární tlačítko „To se mi líbí“, po jehož stisknutí na konkrétní stránce začne model blikat. Uvnitř modelu je opět základní deska Arduino, ke které je připojena deska s LED diodami. Blikání zajišťuje jednoduchý PHP skript, který využívá grafické rozhraní API.



**Obr. 12 Model blikajícího palce [BRANDRICK, CH. 2011]**

## **4.2 Zobrazování dat z webových služeb**

Jako první příklad jsem použil využití dostupných dat z poštovního klienta. Komponenty jsou v tomto případě téměř shodné s předchozím příkladem. Rozdíl je pouze v použití jiné modulové desky.

## Komponenty:

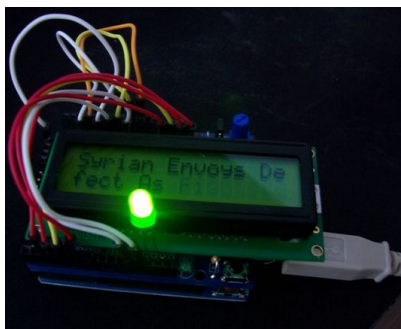
- Arduino Duemilanova (základní modulová deska)
- Arduino Ethernet Shield (síťový modul)
- Arduino LCD Keypad Shield (znakový displej s tlačítky)
- USB kabel (typ A-B)
- UTP kabel (RJ45)

Úloha je opět řešena pomocí PHP skriptu a umožňuje zobrazení počtu doručených emailů ve schránce, zobrazení jména odesílatele a názvu předmětu a možnost smazání vybraných emailů pomocí tlačítka. Potřebná data jsou načítána z emailového serveru POP3.



**Obr. 13 Zobrazení počtu emailů ve schránce  
[ARDUINO EMAIL MANAGER. 2010]**

Dalším příkladem je zobrazování zpráv z informačního RSS kanálu. Hardwarové komponenty jsou zde podobné jako v předchozím případě a samotná realizace je řešena pomocí kódu vytvořeného v programovacím jazyce Python.



**Obr. 14 Zobrazený titulek zprávy z RSS kanálu  
[ARDUINO RSS FEED DISPLAY. 2014]**

## 5 NÁVRH KOMPONENTŮ PRO VLASTNÍ MODEL

V této kapitole jsem nejprve provedl průzkum trhu, popsal a srovnal dostupné komponenty a poté sestavil seznam vhodných hardwarových prostředků pro realizaci vlastního modelu displeje.

### 5.1 Desky a moduly Arduino

Základem každého Arduina je procesor od firmy Atmel, který je doplněn dalšími elektronickými komponenty. Společným prvkem většiny desek je jednotné grafické zpracování. Na většině desek je kromě hlavního čipu také převodník umožňující komunikaci mezi PC (USB) a procesorem.

V následující části jsou uvedeny používané modely desek s popisem základních parametrů a vlastností.

- **Arduino Uno**

Z dlouhodobého hlediska se jedná o zřejmě nejčastěji používaný typ desky. Je přímým následovníkem hlavní vývojové linie, která vznikla vydáním prvního Arduina se sériovým portem místo USB. Dále pak pokračující přes Arduino Extreme, Diecimila a Duemilanove až k současnému modelu Uno. Na desce se nachází procesor ATmega328 a USB.

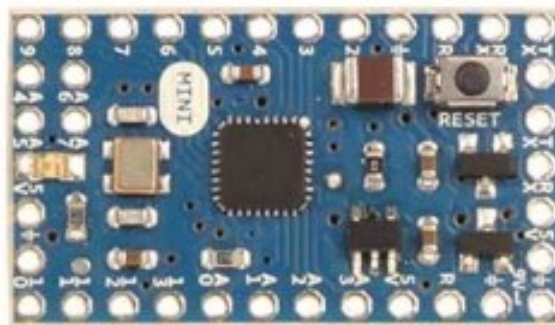


Obr. 15 Arduino Uno [ARDUINO ÚVOD. 2013]



### ▪ Arduino Mini

Nejmenší oficiální verze Arduina. Je navržena především pro úsporu místa. Nevýhodou plynoucí z nedostatku místa je však nepřítomnost USB portu. Pro účely programování je tím pádem nutné použít externí USB 2 Serial převodník. Z hlediska výkonu však nijak výrazně nezaostává za většími modely. Běží na procesoru ATmega328 s taktem 16 MHz. Díky svým malým rozměrům je vhodný pro použití v chytrých vypínačích nebo dálkových ovladačích.



Obr. 16 Arduino Mini [ARDUINO ÚVOD. 2013]

### ▪ Arduino Mega2560

Tento model patří do skupiny desek, jejichž vzhled je výsledkem prodloužení designu Arduina Uno. Lehké zvětšení rozměrů umožňuje lepší využití prostoru pro větší a výkonnější čipy a kromě toho také zapojení více pinů (zdírek). Hodí se zejména v případech, kde je nutný větší výpočetní výkon.



Obr. 17 Arduino Mega2560 [ARDUINO ÚVOD. 2013]



## ▪ Arduino Yún

Tento model je odvozen od základního modelu Arduino Uno. Kromě základního čipu ATmega32u4, na kterém běží jádro Arduina, zde také můžeme najít i čip Atheros AR9331, který je schopen běžet s odlehčenou verzí linuxu Linino. Další výbavou je také softwarový bridge (most), který má za úkol zajištění komunikace mezi oběma čipy. Na desce je k dispozici také Ethernet port a Wifi modul pro možnost připojení k síti. Můžeme tedy posílat a sdílet naměřené hodnoty na webový server.



**Obr. 18 Arduino Yún [ARDUINO ÚVOD. 2013]**

## ▪ Arduino shieldy

Kromě základních desek existuje celá řada připojitelných modulů (shieldů), které jsou s deskami kompatibilní. Připojení se provádí jednoduchým zasunutím pinů do příslušných zdírek na základní desce. Mezi základní dostupné shieldy patří Ethernet shield a Wi-fi shield pro připojení k síti, dále také Motor shield aj.



**Obr. 19 Arduino Ethernet shield [ARDUINO ÚVOD. 2013]**

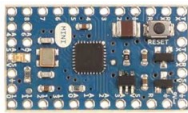


## 5.2 Výběr hardwarových prostředků pro model displeje

Pro realizaci vlastní úlohy jsem se po zvážení a srovnání různých možností rozhodl pro použití základní programovací desky Arduino. Hlavním důvodem byl široký výběr modelů a dobrá dostupnost na českém trhu.

### ▪ Výběr desky

Pro výběr základní desky Arduino jsem zvažoval a srovnával několik kritérií. Hlavními kritérii byla přítomnost Wi-Fi modulu pro bezdrátové připojení k síti a cena. Srovnání možných variant je uvedeno v následující tabulce.

**Tab. 1 Srovnání kritérií pro výběr desky**

	<b>Arduino Mini</b> 	<b>Arduino Uno</b> 	<b>Arduino Yún</b> 
<b>Wi-Fi modul</b>	X	X	✓
<b>Orientační cena</b>	400 Kč	700 Kč	1800 Kč

Nejlevnějším řešením by mohlo být použití desky Arduino Mini. Jedná se o však o nejjednodušší model bez USB portu, což by vyžadovalo použití externího převodníku. Chybí také Wi-Fi modul, který by musel být dokoupen.

Univerzálním řešením by bylo použití desky Arduino Uno, která je v praxi nejpoužívanější a také široce dostupná trhu. Opět by ovšem chyběl požadovaný Wi-Fi modul, který se v českých internetových obchodech dá pořídit za poměrně vysokou cenu.

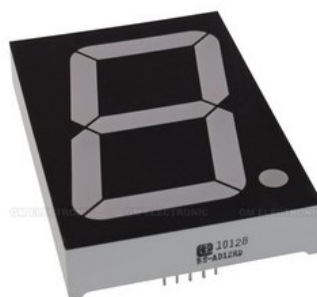
Jako nejlepší řešení se jeví varianta s deskou Arduino Yún. Ze srovnávaných modelů je sice nejdražší, ale má integrovaný Wi-Fi modul a je nejkompaktnější, protože všechny komponenty se nachází na jedné desce. Celková cena je tedy v tomto případě nižší než při použití desky Arduino Uno s externím Wi-Fi modulem.



**Obr. 20 Arduino Yún [ARDUINO ÚVOD. 2013]**

#### ▪ Výběr displeje

Pro účely mého modelu jsem uvažoval použití buď znakového LCD displeje nebo 7-segmentových LED displejů. Univerzálnější a zřejmě jednodušší variantou by byl znakový LCD displej. Důležitým kritériem zde ovšem byly větší požadované rozměry, proto jsem nakonec zvolil 7-segmentové LED displeje. Vybraný typ HD-AD12RD má společnou anodu, výšku znaku 56,9 mm a diody svítí zelenou barvou.



**Obr. 21 7-segmentový LED displej (HD-AD12RD)  
[HD-AD12RD. 2014]**

#### ▪ Výběr ostatních komponent

Pro řízení jednotlivých displejů bude třeba použít ovladače na principu sériový/paralelní. V mém případě jsem zvolil typ TLC5940. Výhodou je dostupná knihovna pro vývojové prostředí Arduino a 16 výstupních kanálů, díky čemuž bude možné k jednomu ovladači zapojit 2 displeje.

Z důvodu vyšší hodnoty napětí a proudového odběru displejů bude třeba model napájet externě. Pro tyto účely jsem vybral 12V/1A napájecí adaptér. Tímto napájecím zdrojem bude, při snížení napětí na 5V použitím lineárního regulátoru, napájena také samotná deska Arduino Yún. Pro rychlejší zapojení a možnost jednoduché pozdější modifikace budu používat nepájivé kontaktní pole. Připojený a zhotovený celek bude poté umístěn do krabičky. Všechny použité komponenty jsou přehledně uvedeny v následující tabulce.

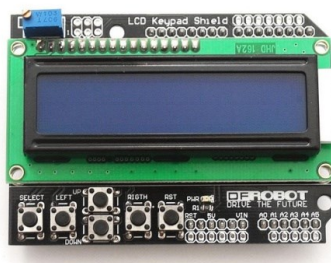
**Tab. 2 Použité komponenty**

KOMPONENT	OZNAČENÍ	POČET KS
Arduino Yún	Arduino Yún	1
7-segmentový LED displej	HD-AD12RD	5
Kabel USB A–micro B (1m)	Premium Cord USB A-micro B	1
Napájecí síťový adaptér (12V/1A)	Vigan VDC	1
Napájecí souosý konektor	PC-2,1	1
Nepájivé kontaktní pole	ZY-204	1
Propojovací kabely M-M/F-M	PJ-20R	70
Posuvný registr TLC5940	TLC5940	3
Rezistor (2k $\Omega$ )	RM 2k	3
Rezistor (10k $\Omega$ )	RM 10k	1
Lineární regulátor napětí (12V/5V)	LM2940CT-5 TO220	1
Keramický kondenzátor (1 $\mu$ F)	CK 1u/50V	1
Elektrolitický kondenzátor (47 $\mu$ F)	CE 47u/35V	1
Plastová konstrukční krabička	KP22 (Z-33)	1

## 6 ZPRACOVÁNÍ ÚLOH ZOBRAZOVÁNÍ DAT

Pro účely realizace vlastní úlohy zobrazování dat bylo nutné se nejprve seznámit s vývojovým prostředím Arduino IDE. Z hlediska uživatelského rozhraní jde o velmi intuitivní a přehledně uspořádané programovací prostředí. Kromě dobře zpracované podpory nabízí také základní příklady s přidanými komentáři.

Na začátek jsem v rámci testování provedl jednoduchou úlohu, pro kterou jsem použil zapůjčený displej LCD Keypad Shield. Tento displej má 2 řádky o 16 znacích a je kompatibilní s vybranou deskou Arduino Yún. Připojení je jednoduché, spočívá pouze v zasunutí shieldu s displejem do příslušných pinů na desce.



**Obr. 22 LCD Keypad Shield**  
**[ARDUINO LCD KEYPAD SHIELD. 2009]**

Pro ověření základních funkcí displeje jsem použil jednoduchý program pro zobrazení zadaného textového řetězce:

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(8, 9, 4, 5, 6, 7);

void setup()
{
    lcd.begin(16, 2);    // počet řádků a sloupců displeje
    lcd.setCursor(0, 0); // nastavení kurzoru na požadovanou polohu
    lcd.print("Test displeje"); // zobrazení zadaných znaků na displeji
    lcd.setCursor(0, 1);
    lcd.print("OK");
}

void loop()
{
}
```

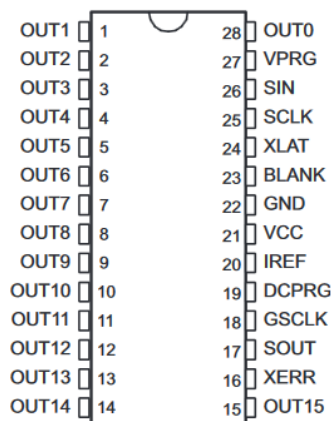


**Obr. 23 Zobrazení zadaného řetězce znaků na displeji**

## 6.1 Sestavení modelu displeje

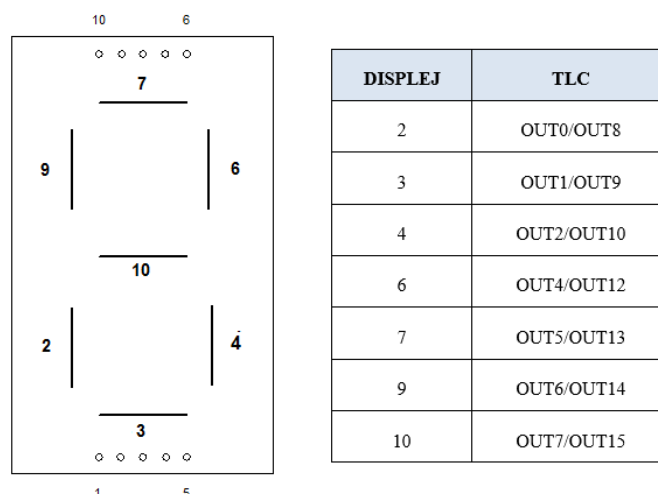
Pro sestavení a následnou montáž celého modelu velkého displeje jsem použil komponenty uvedené v podkapitole 5.2. Celý obvod jsem nejprve schematicky navrhl ve volně dostupném programu Fritzing, podle kterého jsem poté všechny komponenty zapojil.

Pro připojení všech potřebných výstupů jsem použil digitální piny desky Arduino Yún. O řízení jednotlivých displejů se starají ovladače TLC5940, které díky 16 dostupným kanálům umožňují připojení 2 displejů.



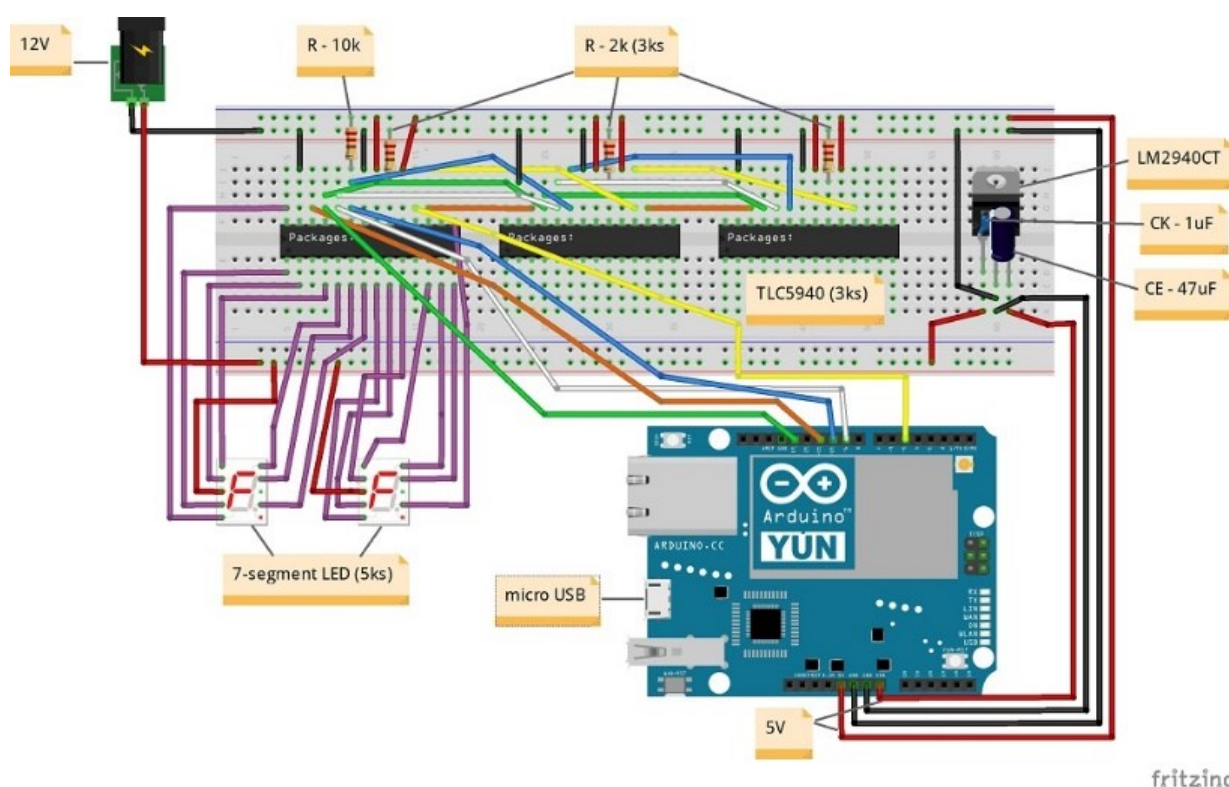
**Obr. 24 Rozložení pinů ovladače TLC5940**  
[TLC5940. 2015]

Ke vstupním pinům ovladače (16-27) jsem přivedl potřebné napájecí a ovládací kontakty a jednotlivé výstupní kanály (OUT0-OUT15) jsem propojil s příslušnými vývody na 7-segmentovém LED displeji. Na následujícím obrázku je uvedeno k jakým kanálům ovladače TLC5940 byly připojeny piny na LED displeji. Piny 1,5 displeje tvoří společnou anodu, ke které bylo přivedeno napájení 12V.



**Obr. 25 Zapojení pinů displeje ke kanálům ovladače TLC5940**

Z důvodu přehlednosti jsem na níže uvedeném schematickém návrhu zapojení obvodu naznačil připojení pouze 2 displejů k ovladači TLC5940. Zbylé 3 displeje jsou postupně připojeny k dalším ovladačům stejným způsobem. Zapojení displejů na obrázku je spíše pro ilustraci a nekoresponduje s reálným zapojením, jelikož v knihovně programu Fritzing přesný typ použitých displejů není dostupný.











**Obr. 26 Schematický návrh zapojení obvodu**



Zapojení všech komponentů na nepájivém kontaktním poli včetně propojení jednotlivých vstupních kanálů ovladačů TLC5940 s příslušnými digitálními piny na desce Arduino Yún je realizováno pomocí propojovacích kabelů. Barevné označení těchto vodičů pro odpovídající typ zapojení je zobrazeno v následující tabulce.

**Tab. 3 Barevné označení kabelů pro odpovídající typ zapojení**

OZNAČENÍ	BARVA
VCC	
GND	
GSCLK	
XLAT	
BLANK	
SIN/SOUT	
GSCLK	
TLC OUT/LED IN	

Takto navržený obvod jsem postupně zapojil a umístil do plastové krabičky, ze které jsem vyvedl konektor micro USB pro připojení desky modelu k PC a konektor pro připojení napájecího adaptéru (12V).



**Obr. 27 Umístění zapojeného obvodu do plastové krabičky**

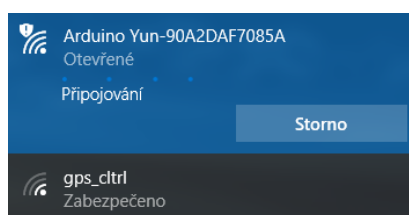




**Obr. 28** Sestavený model displeje a zadní část s vyvedenými konektory

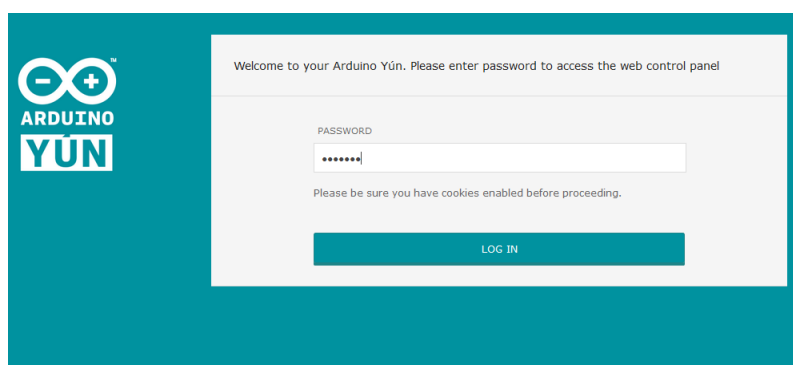
## 6.2 Připojení a konfigurace desky Arduino Yún

Deska Arduino Yún se po zapojení chová jako standardní síťový access point. Nejprve je třeba v seznamu dostupných bezdrátových sítí vyhledat položku pro Arduino Yún a připojit se přímo k desce.



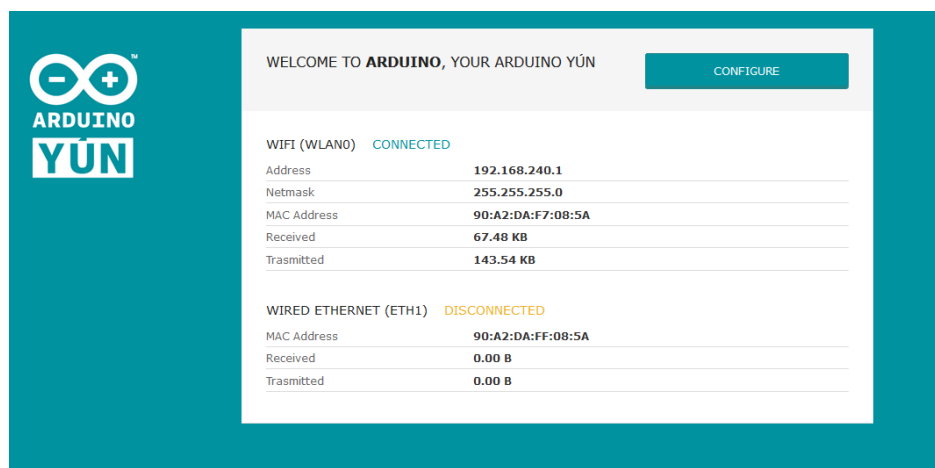
**Obr. 29** Připojení k desce Arduino Yún

Po připojení a zadání výchozí IP adresy (192.168.240.1) do webového prohlížeče se zobrazí jednoduché administrační prostředí. Prvním krokem je zadání výchozího hesla pro administraci („arduino“).



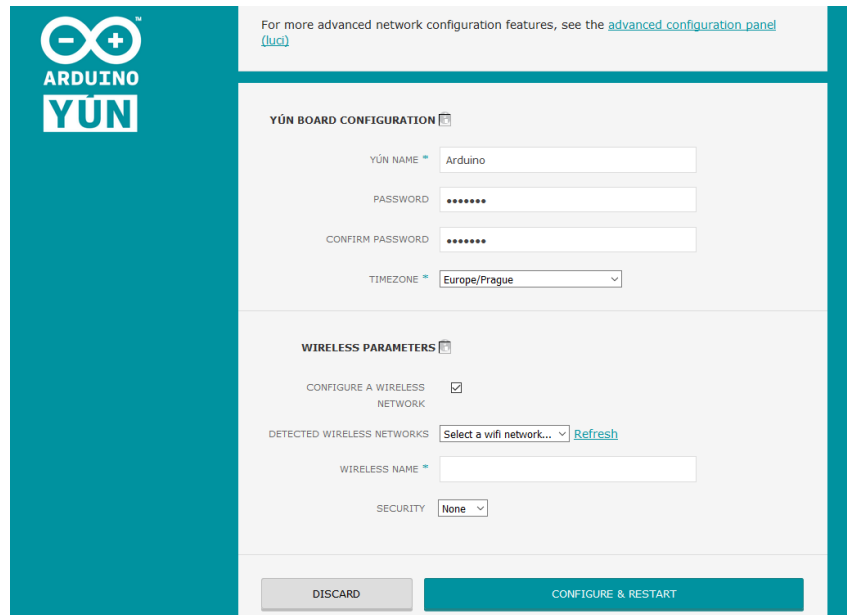
**Obr. 30** Administrační webové prostředí desky Arduino Yún

Pro účely bezdrátové komunikace v rámci Internetu je nutné nastavit připojení k Wi-Fi síti (tlačítko „Configure“).

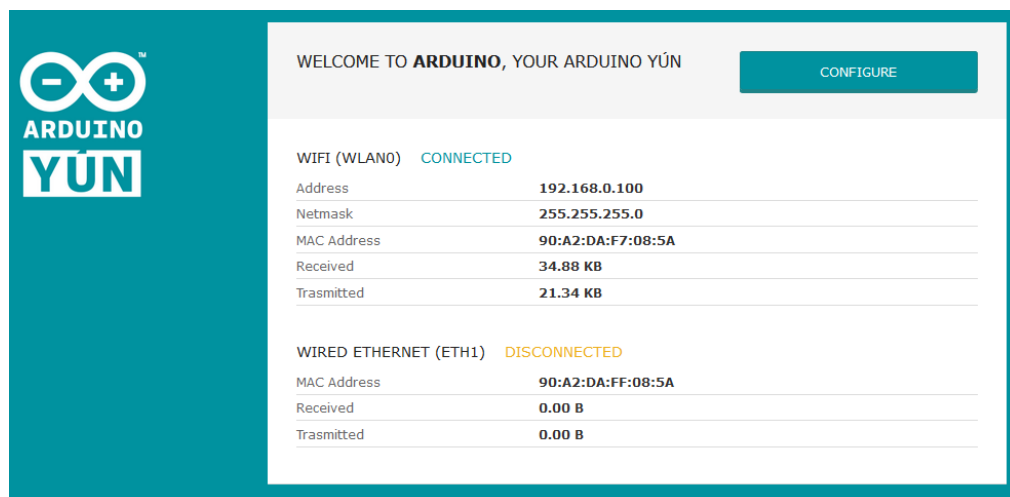


**Obr. 31 Nastavení pro připojení desky k bezdrátové síti**

Ve formuláři pro připojení je třeba vyplnit několik údajů jako je název zařízení, časové pásmo a také nastavení vlastního hesla pro administraci o délce min. 8 znaků. Poté vybereme z nabídky dostupných sítí tu, ke které se bude deska připojovat. Dále zadáme konkrétní bezpečnostní údaje vybrané sítě a po restartu by deska měla navázat komunikaci.



**Obr. 32 Údaje pro nastavení připojení**



Obr. 33 Okno se zobrazením aktuálního stavu připojení desky

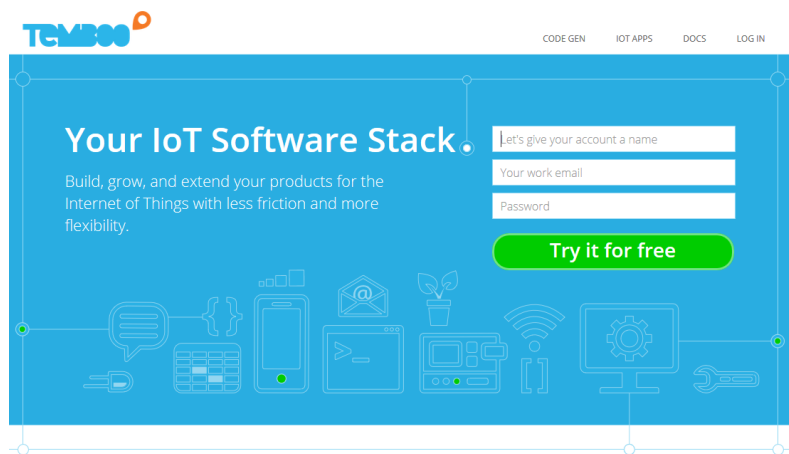
### 6.3 Realizace úloh pro zpracování a zobrazování dat

Princip úloh spočívá ve sběru volně dostupných dat z veřejných webových API a následném zpracování pro výstup ve formě zobrazení číselných hodnot na displeji. Zjednodušené schéma realizace úloh je zobrazeno na následujícím obrázku.



Obr. 34 Princip funkce úloh pro zpracování a zobrazování dat

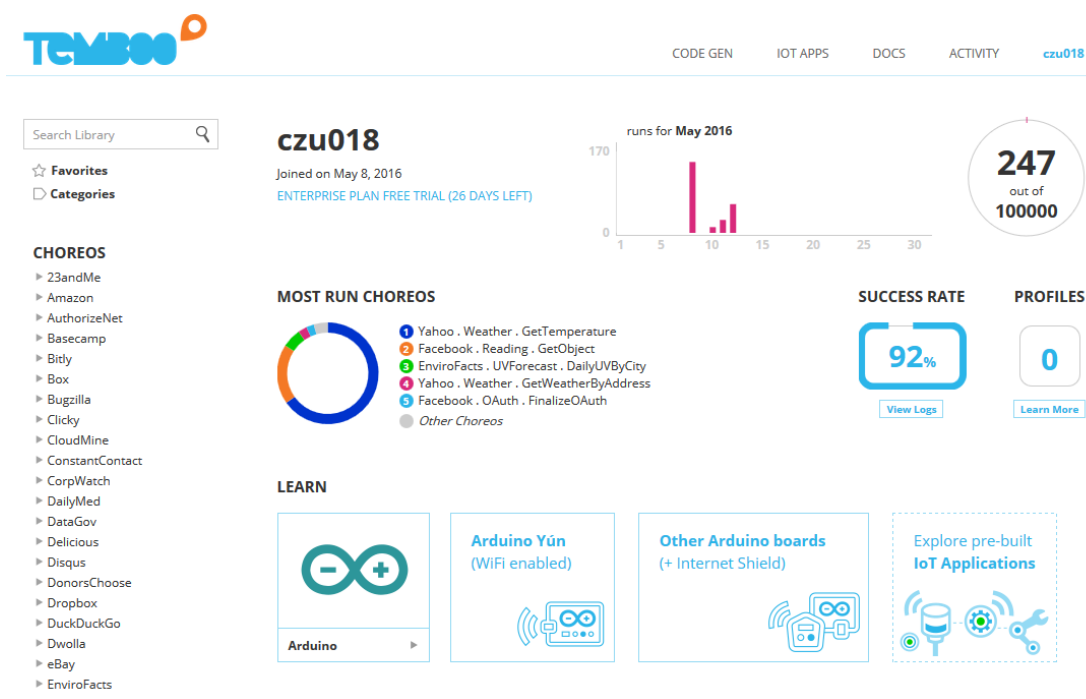
Pro účely čerpání vybraných veřejných dat jsem použil službu Temboo.



Obr. 35 Úvodní stránka webové služby Temboo

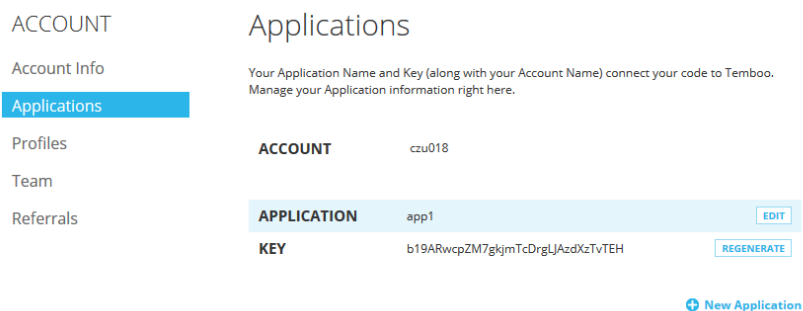
Základním krokem pro používání služby je registrace uživatelského účtu, které probíhá standardně zadáním potřebných kontaktních a přihlašovacích údajů.

Služba v současné chvíli umožňuje zpracování API z více jak 30 zdrojů. Hlavní profilová obrazovka služby zobrazuje kromě informací také statistiky využití a seznam knihoven jednotlivých webových API („Choreos“).



Obr. 36 Profilová stránka uživatelského účtu služby Temboo

Prvním krokem při realizaci úloh bylo vytvoření aplikace („New application“), ke které se přiřadí název a služba vygeneruje aplikační klíč („Application key“). V rámci takovéto aplikace již mohou pokračovat s generováním dat z dostupných API.



**Obr. 37 Vytvořená aplikace (název, aplikační klíč)**

#### ▪ Zobrazování počtu fanoušků univerzitní FB stránky

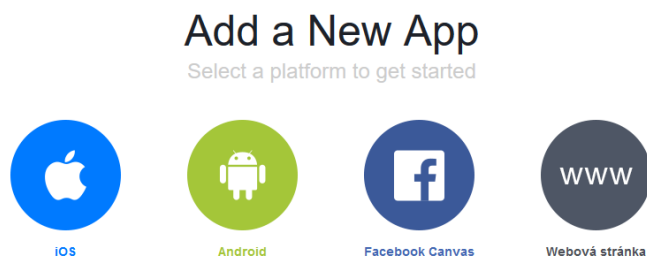
Jako první úlohu zpracování dat jsem zvolil zobrazení počtu fanoušků univerzitního profilu na sociální síti Facebook. Pro získání potřebných dat je třeba zadat vstupní hodnoty, na základě kterých služba provede zpracování a vygeneruje výstup ve formě zdrojového kódu pro Arduino.

Nutným krokem pro získání vstupních dat je vytvoření aplikace v rámci vývojářského účtu na Facebooku (developers.facebook.com). K tomu slouží přihlašovací údaje z klasického uživatelského účtu.



**Obr. 38 Vývojářské prostředí FB účtu**

Pro účely mé úlohy jsem vytvořil novou aplikaci na platformě webové stránky a stejně jako v případě služby Temboo se k názvu aplikace přiřadí unikátní klíč a také ID.



**Obr. 39** Vytvoření nové FB aplikace



**Obr. 40** Přehled údajů o vytvořené aplikaci

Tímto se mohu vrátit do služby Temboo a pokračovat vybráním konkrétní položky pro zpracování FB API. V mém případě jsem pro získání počtu fanoušků potřeboval profilové údaje univerzitní stránky, což znamená, že jsem v seznamu API knihoven vybral možnost Facebook – Reading – GetObject.

Search Library

Arduino Yún

Onboard Ethernet & WiFi

Want to stream sensor data?

Want to use it as a gateway?

Facebook . Reading . **GetObject** ☆

Retrieves the details for a Graph API object that you specify.

Is this Choreo triggered by a sensor event?

**INPUT**

**AccessToken**  
The access token retrieved from the final step of the OAuth process.

**ObjectID**  
The id of a graph api object to retrieve.

**OPTIONAL INPUT**

**Run**

**OUTPUT**

Should an output trigger a hardware event?

**Response**  
The response from Facebook. Corresponds to the ResponseFormat input. Defaults to JSON.

**CODE**

**Obr. 41 Okno nástroje pro načtení profilových dat z FB účtu**

Pro vygenerování potřebného výstupu je jako první potřeba vložit vstupní token („AccessToken“), který systém vygeneruje po provedení autorizačního procesu („OAuth process“). Tento proces zahrnuje 2 kroky („Initialize OAuth“, „Finalize OAuth“).

V rámci prvního kroku je nutné zadat ID FB aplikace („AppID“), kterou jsem předtím vytvořil a údaj s přístupovými právy („Scope“).

### Facebook . OAuth . **InitializeOAuth** ☆

Generates an authorization URL that an application can use to complete the first step in the OAuth process.

Is this Choreo triggered by a sensor event?

#### INPUT

Save Profile

**AppID**

The App ID provided by Facebook.

1698485770415210

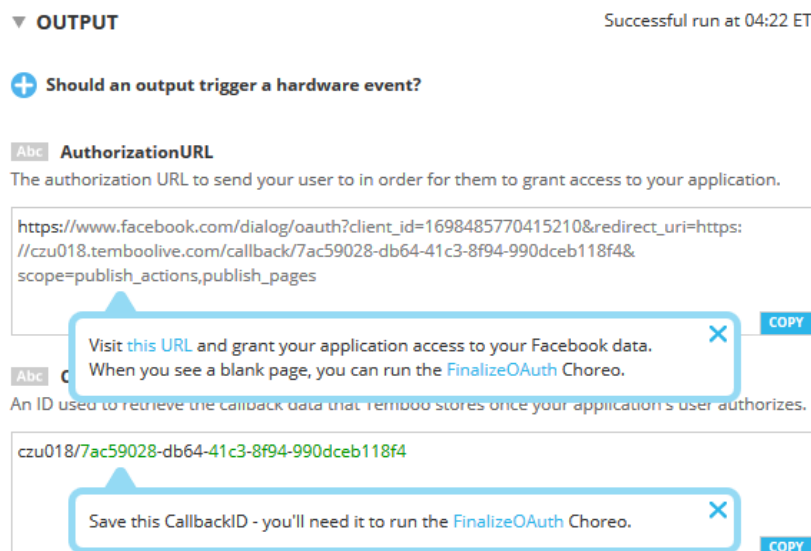
**Scope**

A comma-separated list of permissions to request access for (i.e. "publish\_actions,read\_mailbox"). For more information see Choreo notes.

publish\_actions,publish\_pages

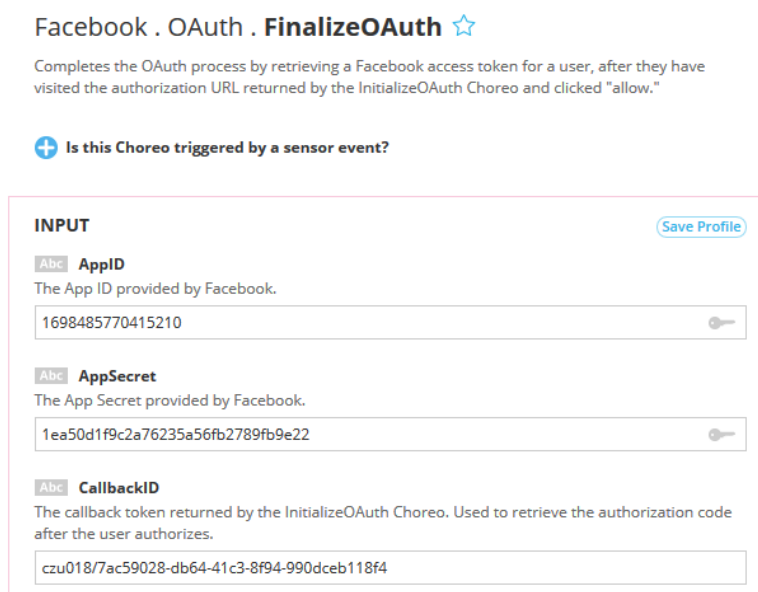
**Obr. 42 První krok autorizačního procesu**

Po potvrzení údajů se vygenerují výstupní údaje, které slouží pro další krok autorizačního procesu. Prvním výstupním údajem je URL stránky („AuthorizationURL“), kterou je třeba otevřít a tím se krok potvrdí. Druhým výstupním údajem je unikátní ID („CallbackID“), který slouží jako vstupní údaj pro druhý krok autorizačního procesu.



Obr. 43 Výstupní údaje prvního kroku autorizačního procesu

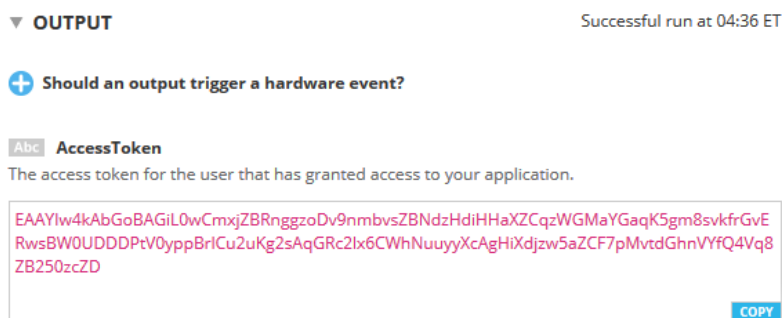
Nyní již mohu přejít ke druhému kroku autorizačního procesu („Finalize OAuth“) a zadat všechny potřebné údaje.



Obr. 44 Zadání vstupních údajů pro druhý krok autorizačního procesu

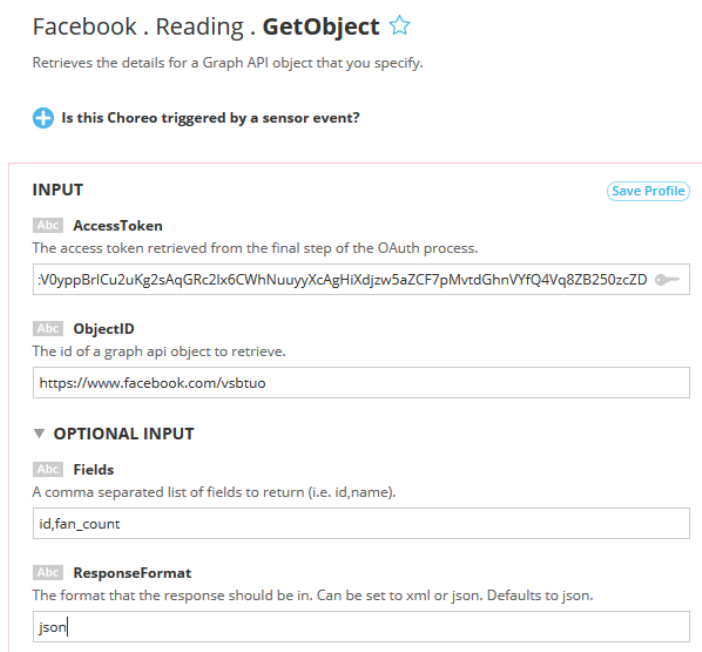
Po dalším potvrzení se vygeneruje vstupní token („AccessToken“), který již mohu zadat pro samotné vygenerování zdrojového kódu pro Arduino.





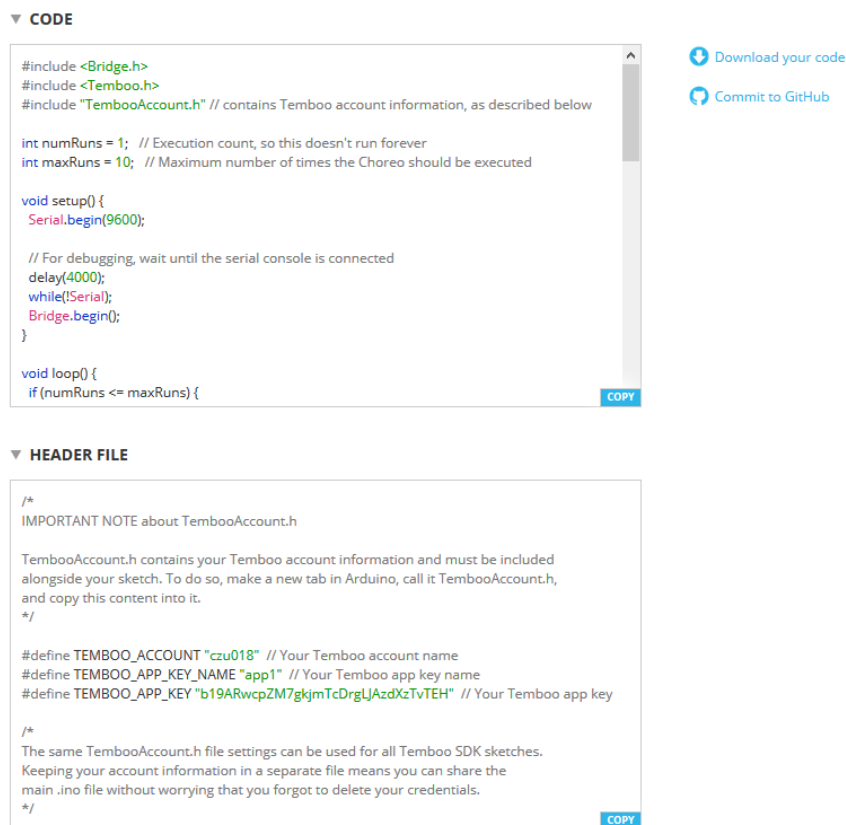
**Obr. 45 Vygenerovaný vstupní token**

Po návratu na stránku pro načtení dat z FB je třeba kromě zmíněného vstupního tokenu zadat ještě adresu FB stránky, ze které budu data čerpat (univerzitní FB stránka), požadované údaje pro načtení (ID stránky, počet fanoušků) a formát výstupních dat (json).



**Obr. 46 Zadání všech potřebných dat pro vygenerování zdrojového kódu pro Arduino**

Tímto je proces ze strany služby Temboo hotový a je možné stáhnout připravený zdrojový kód do PC.



**Obr. 47 Zdrojový kód pro stažení do PC**

Po otevření zdrojového kódu ve vývojovém prostředí Arduino IDE je třeba provést kontrolu všech dat. Kód se skládá z hlavního souboru a hlavičkového souboru s údaji o aplikaci, které vygeneruje služba Temboo.



**Obr. 48 Hlavičkový soubor s údaji o aplikaci Temboo**

Vygenerovaný kód slouží jako základní část pro další úpravy, které je třeba provést pro požadovanou výslednou podobu kódu. V první fázi bylo třeba vybrat požadovanou datovou položku, jelikož se data na výstupu zobrazují ve formátu json. V případě úlohy pro zobrazování počtu fanoušků univerzitní FB stránky se jedná o položku „likes“. Pro další práci s touto hodnotou jsem převedl datový typ na celočíselný formát (int) a mohl jsem tak definovat novou proměnnou (long int fans), ve které se bude ukládat požadovaná hodnota počtu fanoušků. V této fázi je již možno provést kompilaci a nahrání programu do připojené

desky Arduino Yún. Pro kontrolu výstupních údajů je nutné otevřít sériový monitor a informace zkontrolovat.



**Obr. 49 Hodnota počtu fanoušků na výstupu sériového monitoru  
vývojového prostředí Arduino**

Dalším bodem byla úprava kódu pro možnost zobrazení hodnoty na displeji sestaveného modelu. V této části jsem postupně definoval jednotlivé segmentové číslice a pomocí několika přidáných proměnných a funkcí jsem nakonfiguroval finální podobu zdrojového kódu. Výstupem je požadovaná číselná hodnota počtu fanoušků univerzitní FB stránky.



**Obr. 50 Zobrazená hodnota počtu fanoušků univerzitní  
FB stránky (7732)**

#### ▪ Zobrazování hodnoty aktuální teploty

Jako druhou úlohu jsem vybral zobrazování aktuální teploty podle zadaného města. Služba Temboo k tomu využívá API webové meteorologické služby Yahoo Weather (Yahoo – Weather – GetTemperature). V tomto případě je postup pro vygenerování dat jednodušší, protože vstupním údajem je pouze název města a jednotka teploty. Výstupem je opět jako v minulém případě hodnota ve formátu json.

Yahoo . Weather . **GetTemperature** ☆

Retrieves the current temperature from Yahoo Weather for the specified location.

+ Is this Choreo triggered by a sensor event?

**INPUT** [Save Profile](#)

**Address**  
The address to be searched.  
Ostrava

**OPTIONAL INPUT**

**Units**  
The unit of temperature in the response. Acceptable inputs: f for Fahrenheit or c for Celsius. Defaults to f. When c is specified, all units measurements returned are changed to metric.  
C

**Run**

**OUTPUT** Successful run at 13:55 ET

+ Should an output trigger a hardware event?

**Temperature**  
The current temperature (defaults to Fahrenheit).  
22 [COPY](#)

**Obr. 51 Zadání vstupních dat a vygenerování výstupu pro zobrazení aktuální teploty**

Stejně jako v první úloze jsem stažený základní kód upravil a hodnotu teploty převedl na datový typ (int) s definovanou proměnnou (long int temp). Po úspěšné kompilaci a nahrání programu na desku mohu zkontrolovat výstupní data.



**Obr. 52 Údaj o aktuální teplotě na výstupu sériového monitoru ve vývojovém prostředí Arduino**

Následná úprava kódu pro zobrazení hodnoty teploty na displeji modelu je v tomto případě velice podobná jako v minulé úloze. Přidal jsem navíc příkazy pro zobrazení symbolu a stupně Celsia.



Obr. 53 Zobrazení hodnoty teploty (22°C)

#### ▪ Zobrazení hodnoty UV indexu

Poslední úlohou je zobrazení hodnoty UV indexu podle zadaného města. Zde služba Temboo využívá data z API služby EnviroFacts. Jelikož se jedná o americkou službu, lze načíst údaje pouze pro města v USA.

Postup pro vygenerování údajů je v tomto případě podobný jako při zobrazení teploty. Vstupními údaji jsou název města, stát, ve kterém se nachází a formát výstupních dat.

EnviroFacts . UVForecast . **DailyUVByCity** ☆

Retrieves EPA daily Ultraviolet (UV) Index readings in a given city.

+ Is this Choreo triggered by a sensor event?

**INPUT** [Save Profile](#)

**City**  
A valid City Name in the United States.

**State**  
The abbreviation of the state that the city resides in.

**OPTIONAL INPUT**

**ResponseType**  
Results can be retrieved in either JSON or XML. Defaults to XML.

[Run](#)

**OUTPUT** Successful run at 07:08 ET

+ Should an output trigger a hardware event?

**Response**  
The response from EnviroFacts.

```
[{"CITY":"BOSTON","STATE":"MA","UV_INDEX":3,"UV_ALERT":0}]
```

[COPY](#)

Obr. 54 Zadání vstupních dat a vygenerování výstupu pro zobrazení UV indexu

Vygenerovaný zdrojový kód pro stažení opět vyžaduje úpravu a převod hodnoty na datový typ (int) s definovanou proměnnou (long int uv). Po kompilaci a nahrání programu můžeme na výstupu sériového monitoru zkontrolovat požadovanou hodnotu.



**Obr. 55 Hodnota UV indexu na výstupu sériového monitoru ve vývojovém prostředí Arduino**

Posledním krokem je vykonání úprav kódu nutných pro zobrazení hodnoty na displeji modelu.



**Obr. 56 Zobrazení hodnoty UV indexu (3)**

Stažené základní zdrojové kódy a upravené kódy s přidanými komentáři pro všechny 3 realizované úlohy jsou k dispozici na přiloženém CD. V příloze se kromě toho nachází také soubor pro instalaci vývojového prostředí Arduino, knihovna pro ovladače TLC5940 a také kód pro reset wi-fi modulu desky Arduino Yún.

## 7 ZÁVĚR

Cílem diplomové práce bylo využití prostudovaných a získaných znalostí z oblasti open-source hardware a webových služeb pro následný návrh a realizaci vlastního řešení v podobě modelu displeje pro zpracování a zobrazování veřejně dostupných dat. Práce je rozdělena do několika kapitol, které se věnují jak obecnému popisu a základním informacím, sloužícím jako základ pro pochopení problematiky, tak i konkrétním krokům při řešení praktických úloh.

Teoretická část práce měla za úkol seznámení se ze základními informacemi a možnostmi open-source hardware. Byly popsány a vysvětleny pojmy týkající se tohoto tématu a dále uvedeny konkrétní příklady těchto zařízení. Následující kapitola je zaměřena na webové služby a API. Zde jsem opět přiblížil základní informace, které se týkají tohoto tématu, přidal příklady webových služeb s ohledem na přístup k API a také stručně popsal formáty a protokoly, které slouží pro výměnu dat. V další kapitole jsem se věnoval příkladům realizovaných řešení, která využívají open-source hardware pro zobrazování veřejných dat z webových rozhraní API. Vybral jsem a krátce popsal několik případů použití a prezentace veřejných dat ze sociální sítě Facebook a z webových služeb (poštovní klient, informační RSS kanál).

V praktické části práce jsem postupně navrhl a realizoval vlastní model pro zpracování a zobrazování vybraných dat z veřejných API. Nejprve jsem provedl průzkum trhu, na základě kterého jsem poté vybral vhodné komponenty pro sestavení modelu displeje. Model je založen na základní programovací desce Arduino Yún. Hlavní výhodou tohoto modelu je integrovaný wi-fi modul pro bezdrátovou komunikaci. Jedním z hlavních požadavků pro zobrazení dat byla velikost displeje. Zvolil jsem proto 7-segmentové LED displeje s většími rozměry. Pro realizaci úloh sběru a zpracování dat z API vybraných webových zdrojů jsem použil službu Temboo. Pomocí této služby jsem získal data a stáhnul základní zdrojové kódy pro další zpracování. Jako zdrojová data pro zobrazování jsem zvolil počet fanoušků univerzitní stránky na Facebooku, aktuální teplotu a hodnotu UV indexu. Pro všechny úlohy jsem vytvořil zdrojové kódy pro zobrazení požadovaných číselných hodnot na displeji modelu.

## SEZNAM POUŽITÝCH ZDROJŮ

Arduino Email Manager. [online]. [cit. 2016-04-20]. Dostupné z: <http://opensource-torchris.blogspot.cz/2010/05/arduino-email-manager-part-1.html>

Arduino LCD Keypad Shield [online]. [cit. 2016-04-20]. Dostupné z: <http://www.droboticsonline.com/index.php/arduino-lcd-keypad-shield.html>

Arduino RSS Feed Display. [online]. [cit. 2016-04-20]. Dostupné z: <http://www.instructables.com/id/Wiring-up-the-LCD-and-the-LED/?ALLSTEPS>

Arduino Úvod. [online]. [cit. 2016-04-20]. Dostupné z: <http://www.hwkitchen.com/news/arduino-uvod-1/>

Arduino YUN. [online]. [cit. 2016-04-20]. Dostupné z: <http://www.alza.cz/arduino-yun-d569253.htm#popis>

BECHYNSKÝ, Štěpán. .NET Gadgeteer – FEZ Spider. [online]. [cit. 2016-04-20]. Dostupné z: <http://blogs.msdn.com/b/vyvojari/archive/2012/01/16/net-gadgeteer-fez-spider.aspx>

BRANDRICK, Chris. Facebook Likes This: Lego-Arduino Hack Shows Your Likes. [online]. [cit. 2016-04-20]. Dostupné z: [http://www.techhive.com/article/222196/facebook\\_likes\\_this\\_lego\\_arduino\\_hack\\_shows\\_your\\_likes.html](http://www.techhive.com/article/222196/facebook_likes_this_lego_arduino_hack_shows_your_likes.html)

DRYJÁK, T. Tvorba výukového modulu v rámci sociálních sítí. VŠB Technická univerzita Ostrava. Bakalářská práce. 2012

Facebook.com. Facebook for Developers [online]. 2014 [cit. 2016-04-20]. Dostupné z: <http://developers.facebook.com>

Facebook Like box. [online]. [cit. 2016-04-20]. Dostupné z: <http://www.skolti.com/lab/exp8/eng/>

FUS, Alois. Internet věcí a systémy reálného času. VŠB Technická univerzita Ostrava. Bakalářská práce. 2011.

HD-AD12RD. [online]. [cit. 2016-04-20]. Dostupné z: <https://www.gme.cz/hd-ad12rd-p512-093>



HLAVÁČEK, J. Facebook API. Vysoká škola ekonomická v Praze. Bakalářská práce. 2011.

HOLLISTER, Sean. Intel announces Edison, a computer the size of an SD card. [online]. [cit. 2016-04-20]. Dostupné z: <http://www.theverge.com/2014/1/6/5282472/intel-announces-edison-a-computer-the-size-of-an-sd-card>

HORÁK, Milan. Raspberry Pi: Co to vlastně je. [online]. [cit. 2016-04-20]. Dostupné z: <http://www.abclinuxu.cz/clanky/raspberry-pi-co-to-vlastne-je>

Internet věcí. [online]. [cit. 2016-04-20]. Dostupné z: [http://cs.wikipedia.org/wiki/Internet\\_v%C4%9Bc%C3%AD](http://cs.wikipedia.org/wiki/Internet_v%C4%9Bc%C3%AD)

LAVIN, Peter. PHP - objektově orientované: koncepty, techniky a kód. 1. vyd. Praha: Grada, 2009, 211 s. ISBN 9788024721378.

MARGOLIS, Michael. Arduino cookbook. 2nd ed. Sebastopol, Calif.: O'Reilly, c2012, 699 p. ISBN 1449313876.

MELGAR, Enrique Ramos, Ciriaco CASTRO DÍEZ a Przemek JAWORSKI. Arduino and Kinect projects: Design, build, blow their minds. New York: Distributed to the book trade worldwide by Springer Science+Business Media, c2012, 393 p. ISBN 9781430241683.

Microsoft .NET Gadgeteer Simplifies Building Gadgets. [online]. [cit. 2016-04-20]. Dostupné z: <http://news.softpedia.com/news/Microsoft-NET-Gadgeteer-Simplifies-Building-Devices-215655.shtml>

OLŠAN, Jan. Intel uvedl modul Edison pro internet věcí: mikro PC s WiFi, Atomem i Quarkem. [online]. [cit. 2016-04-20]. Dostupné z: <http://www.cnews.cz/intel-uvedl-modul-edison-pro-internet-veci-mikro-pc-s-wifi-atomem-i-quarkem>

OSHOWA. [online]. [cit. 2016-04-20]. Dostupné z: <http://www.oshwa.org/wp-content/uploads/2013/10/what-is-open-source-hardware.jpg>

POLÁK, Michal. Hardware a open source. [online]. [cit. 2016-04-20]. Dostupné z: <http://www.abclinuxu.cz/clanky/hardware-a-open-source>

Raspberry Pi. [online]. [cit. 2016-04-20]. Dostupné z: [http://en.wikipedia.org/wiki/Raspberry\\_Pi](http://en.wikipedia.org/wiki/Raspberry_Pi)

RUBEŠ, Petr. Využití akcelerometru v mobilním telefonu k řízení vzdáleného zařízení. VŠB Technická univerzita Ostrava. Diplomová práce. 2013.

RYBA, Albert. Open source hardware: Ve znamení nových alternativ. [online]. [cit. 2016-04-20]. Dostupné z: <http://www.ictmanazer.cz/2011/11/open-source-hardware-ve-znameni-alternativ/>

Smiirl. [online]. [cit. 2016-04-20]. Dostupné z: <https://www.facebook.com/smiirl>

SOCHOROVÁ, M. Studie a zhodnocení zahraničních volně dostupných API mapových služeb. VŠB Technická univerzita Ostrava. Diplomová práce. 2008

ŠŤASTNÝ, Jakub. Open source hardware: Budoucnost programovatelných hradlových polí. [online]. [cit. 2016-04-20]. Dostupné z: <http://www.scienceworld.cz/technologie/open-source-hardware-budoucnost-programovatelnych-hradlovych-poli-2109/>

Temboo [online]. [cit. 2016-04-20]. Dostupné z: <https://temboo.com/>

ThingSpeak [online]. [cit. 2016-04-20]. Dostupné z: <https://thingspeak.com/>

TLC5940 16-Channel LED Driver[online]. [cit. 2016-04-20]. Dostupné z: <http://www.ti.com/lit/ds/symlink/tlc5940.pdf>

.